Fall 2004 (RT-04F) Rich Transcription Evaluation Plan

1 INTRODUCTION

The goal of this document is to define the evaluation tasks, performance measures, and test corpora to support the 2004 Rich Transcription (RT-04F) fall evaluation. Rich Transcription (RT) is broadly defined to be a fusion of speech-to-text (STT)[1] technology and **m**eta**d**ata **e**xtraction (MDE) technologies which will provide the basis for the generation of more usable transcriptions of human-human speech for both humans and machines. The RT-04F evaluations will support DARPA's Effective, Affordable, Reusable Speech-to-text (EARS) Program.[2] In addition to EARS contractors, these evaluations are open to all interested volunteers. Evaluation will be supported for ten tasks:

Speech To Text (STT) tasks

- Unlimited time STT
- Less than or equal to twenty times realtime STT
- Less than or equal to ten times realtime STT
- Less than or equal to one times realtime STT

Metadata Data Extraction (MDE) tasks

Structural Metadata

- Edit Word Detection
- Filler Word Detection
- IP Detection
- SU Boundary Detection

Diarization

- Who spoke when

Integrated STT and Metadata Tasks

- Speaker Attributed STT
- 04 Rich Transcription

The RT-04F STT evaluations will be on English, Mandarin, and Arabic data while the RT-04F Metadata evaluations will be limited to English language only.

1.1 PRIMARY VS. CONTRASTIVE SYSTEMS

Primary systems: Participants must submit output from exactly one *primary* system[3] for each task they participate in. The primary system must be run on the speech-input condition (see section 9) and can also be run on other conditions[4] specified in section 9. Only the primary systems will be compared across sites.

Contrastive systems: Participants may submit output from additional *contrastive* systems, for tasks on which they have submitted output from a primary system. But each contrastive system must also be run on the required conditions[5]. These contrastive system submissions will only be used for intra-site comparisons.

Additional required condition for EARS STT Contractors: For the STT tasks, EARS contractors must make a primary submission on the Eval-04 data set, and a corresponding submission from the *same system* on the progress test set. Participants who are not EARS contractors will not run the progress test set at all.

1.2 CHANGES FROM RT-03

This section briefly lists the differences between the RT-03S and RT-03F evaluations and RT-04.

1.2.1 CHANGES FROM RT-03S

- The RT-04F STT evaluation will have no significant changes from the STT evaluation in RT-03S.
- Submissions for the speaker diarization evaluation (who spoke when) will be in RTTM format, rather than MDTM.
- There will be no required processing speed task.
- The data and data sources will be new.

1.2.2 CHANGES FROM RT-03F

- There will be only one official set of metrics (the ones defined in this evaluation plan) and one official data format (RTTM). NIST will provide scoring software implementing this evaluation plan. The "BBN Rich Transcription Framework" is no longer included in this evaluation plan.
- Subtypes of filler words and SU's will be evaluated.
- The metrics and scoring for the Speaker Attributed STT task will only count a token as correct if the token is correct for STT purposes and also has the speaker correct. SASTT will not apply to CTS.
- The UEM files will be substantially changed and will focus on defining the data to be processed. Exclusions of small regions (e.g., around speaker-attributed non-speech sounds, unannotated SUs) will be done by the scoring software rather than via UEM files.
- In the RTTM specification (Appendix A) we have renamed **propername** to **propernoun** and renamed

[1] formerly known as automatic speech recognition (ASR)

[2] The EARS research effort is dedicated to developing powerful new speech transcription technology that provides substantially richer and more accurate transcripts than are currently possible. The research focus is on natural, unconstrained speech from broadcasts and telephone conversations in a number of languages. The program objective is to create core enabling technology suitable for a wide range of advanced applications.

[3] That submission is to be designated as primary — see the description of the SYSID string in section 10.3.1.

[4] Those submissions will still be *primary*.

[5] That submission will still be *contrastive* not *primary*.

lip-smack to **lipsmack**, in order to correspond to actual practice and to actual reference data.

- The data and data sources will be new.

2 BACKGROUND

While the traditional STT evaluations have provided a mechanism for evaluating word accuracy, it is clear that words alone are insufficient to formulate a transcription of speech that is maximally useful. A verbatim transcription of the speech stream into a string of lexical tokens yields a transcript that is often difficult to understand. This is because spoken language is much more than just a string of lexical tokens. It contains information about the speaker, prosodic cues to the speaker's intent, and much more. Spoken language also contains disfluencies, which speakers correct and which textual renderings should delete. All of this makes the task of rendering spoken language into text a great challenge, especially with less-than-perfect automatic speech recognition (ASR) performance.

Beginning in the early 1980's, evaluation of ASR stabilized on the current performance measure of word error rate (WER). This measure scores ASR performance using a case-less lexicalized form of ASR output known as the Standard Normalized Orthographic Representation (SNOR) format.[6] The WER is defined as the sum of all ASR output token errors divided by the number of scoreable tokens in a reference transcription of the test data. There are three types of errors: tokens that are missed (deletion errors), inserted (insertion errors), and incorrectly recognized (substitution errors).[7]

Transcripts with the sorts of metadata called for by the RT-04F evaluations will be easier for humans to read and can be processed in more useful ways by computers. The EARS program has chosen to focus metadata extraction efforts on the goal of supporting the creation of transcripts that are more readable and more understandable for the reader.

Solving these problems is the challenge that the EARS program takes as its objective and what the RT evaluation series seeks to assess – namely to develop technology that transforms spoken language into a form that is maximally informative. This requires new approaches to acoustical modeling and insightful models of disfluencies, dialogue and other relevant speaker behaviors. The EARS program has an overarching goal of making large improvements in STT accuracy, and it is expected that the metadata extraction aspects of the program will also advance that goal.

2.1 THE NATURE OF DISFLUENCIES (IN BRIEF)

Spoken disfluencies are portions of speech in which a speaker's utterance is not complete and fluent: speech that the speaker corrects, repeats, or abandons.

Although the full form of the common structure to edit disfluencies is not always present, it occurs in some edit disfluencies and is described in this paragraph. The full form begins with the speaker's fluent initial attempt at an utterance followed by a prosodic transition from fluent to non-fluent speech. The initial attempt is known as the *reparandum* and is followed by an *interruption point*. Next in the full form comes an *editing phase* (sometimes called the *editing phrase*) consisting of *fillers* (words that act as pause fillers, discourse markers, or explicit editing terms). The full form of an edit disfluency ends with a *repair* (which we will call a *correction*) — a repetition or corrected version of the reparandum.

We have three types of edit disfluencies: repairs, repetitions, and restarts. They are defined as follows.

An edit disfluency in which the *correction* is a corrected version of the *reparandum* is an edit disfluency of type *repair*.

An edit disfluency in which the correction repeats the reparandum is classified as an edit disfluency of type *repetition*.

Some edit disfluencies do not have the full form. Any type of edit disfluency may have and empty editing phase (no editing phase). An edit disfluency of type *restart* has a reparandum but no related correction (the speaker simply abandons what they were saying in the reparandum). But there is an interruption point in every edit disfluency, at the [right-hand] end of the reparandum.

Distinguished from[8] edit disfluencies, a filler disfluency (or simply "filler") consists of an interruption point followed by one for more filler words. The interruption point is thus at the beginning of the filler disfluency. There are four subtypes of fillers defined by the Simple Metadata Annotation Specification:

- pause fillers,
- discourse markers,
- explicit editing terms (in the editing phase of an edit disfluency), and
- asides or parentheticals[9] (which are not evaluated).

Disfluencies may occur in succession, and disfluencies of any type may nest inside disfluencies of any type. Edit disfluencies nesting inside other edit disfluencies create *complex disfluencies*. This is quite common, however the complications that this creates will be glossed over in RT-04F and in the reference data annotation.

2.2 THE RT-04F MODEL OF DISFLUENCIES

Because the metadata annotation of the reference data is an expensive (labor intensive) process, the model of spoken disfluencies in the preceding section is simplified in the RT-04F evaluation.

[6] Since some languages' written forms are not word-based, this concept has been extended to cover lexemes – a representation of a written unit of meaning within a language. Thus, this document frequently refers to lexemes, lexical tokens, or tokens rather than words. For English, these terms may be treated more or less equivalently.

[7] Underlying the tabulation of errors is a requirement to align the tokens in the system output transcript with the tokens in the reference transcript. Traditionally, this has been done using a dynamic programming algorithm that searches for an alignment that minimizes the WER.

[8] Fillers and edits are not totally independent, since the editing phase of an edit (if present) is a filler. But fillers also occur by themselves.

[9] Asides and parentheticals are treated as one subtype in SimpleMDEV5.0 and are not evaluated in the RT-04F evaluation.

RT-04F will not include any treatment of the *correction* portion of edit disfluencies—in fact the [right-hand] end of the *correction* will not even be marked in the annotation of the reference data.

Although edit disfluencies are often nested; the EARS program has decided to address only the top-most level of these *complex disfluencies* at this time, and prohibit annotation as nested edit disfluencies. If the original reparandum has multiple serial adjacent disfluencies[10], then the annotated AG file (the reference data format actually produced by the Linguistic Data Consortium) will indicate multiple deletable regions, with an IP at the [right-hand] end of each. Similar treatment (as multiple deletable regions) occurs with, for example, a repetition nested inside a restart[11] In many other cases of complex disfluencies, however, the complex disfluency will be annotated as a series of simple adjacent disfluencies, rather than as one disfluency with multiple interruption points.

The RT-04F model of disfluencies is more fully discussed and explained in the Simple Metadata Annotation Specification[12]. The disfluency task that systems are to perform in RT-04F is to identify the regions that are annotated (following the Simple Metadata Annotation Specification) as deletable.

The two disfluency types, edits and fillers, are independent speech events, although they have similar structure. Thus, their detection has been divided into separate tasks. In RT-04F (as in RT-03) the editing phase of an edit disfluency is treated as a filler disfluency in its own right—thus, the deletable region of a simple edit disfluency (the *reparandum*) is followed by a filler (the *editing phase*) that is also deletable and is evaluated separately.

2.3 DEFINITION OF "DELETABLE REGIONS"

As was the case in RT-03, the metadata extraction research in RT-04F is intended to support the creation of transcripts with disfluencies "cleaned up" and with capitalization and punctuation associated with the sentence-like units. The cleanup will include deleting parts of disfluencies. The *deletable region*[13] of a simple edit disfluency is the time taken by the reparandum. The entire time taken by a filler disfluency is deletable. In a complex [edit] disfluency, the deletable region is as annotated following the SimpleMDE Annotation Specification and may include some fillers that are annotated as part of the reparandum. Evaluation will include a focus on the systems' ability to identify the regions of time that contain deletable regions of disfluencies.[14]

The reader should keep in mind that "*deletable region*" is not meant to imply a structural part of a single disfluency, but rather a stretch of time during which a sequence of words is uttered.

3 THE RT-04F SPEECH TO TEXT (STT) TASKS

Speech To Text systems will be evaluated separately from other submissions. There are three STT processing speed tasks:

Unlimited time (sttul)

Less than or equal to twenty times realtime (stt20x),

Less than or equal to ten times realtime (stt10x), and

Less than or equal to one times realtime (stt1x).

Participants can build systems for any of the listed processing speeds. However, EARS contractors are required to meet the error rate goals based on processing speed. The RT-03F error rate targets are based on stt10x broadcast news systems and stt20x conversational telephone speech systems. EARS contractors are expected to submit systems with these processing speeds.

3.1 DEFINITION OF THE STT PROCESSING SPEED TASKS

The three processing speed tasks are defined as the ratio of the wall-clock Total Processing Time (TPT) divided by the duration of the recorded audio input. TPT is defined (see Appendix B) as the time it takes to process all channels of the recorded speech[15] (including ALL I/O) on a single CPU. The TPT does not include echo cancellation time, time between batch processes, or system start-up time (e.g., booting up and loading initial default models into memory). To elaborate, systems that are not completely pipelined should not count time in between batch processes. Further, some of the data will be distributed without echo cancellation, in order to allow participants to use the echo cancellation algorithms of their choice. Participants will not necessarily pipeline their echo cancellation to run at the same time as their other processing, so to keep the playing field level, echo cancellation time does not count as part of the TPT.

The system description for each STT submission should include processing time information, calculated as described in Appendix B.

3.1.1 ECHO CANCELLATION

Some evaluation test material is distributed without echo cancellation, so that systems may use the echo cancellation algorithm of their choice. The algorithm that NIST has traditionally used in preparing training and test material in the past is the echo cancellation software available from the Mississippi State archive:

http://www.isip.msstate.edu/projects/speech/software/legacy/fir_echo_canceller/index.html

3.2 SCOREABLE STT TOKENS

The existing scoring conventions will be used unchanged (in particular, they will be the same as in RT-03S). RT-04F will score lexical tokens and will not score non-lexical speaker sounds

[10] For example, "**Yeah** but [the * the big * the b- * the big] * **um** the betrayal or whatever she called it." But the annotation tool *cannot* output a reparandum with multiple IP's (internal IP's).

[11] For example, "[That is better than * than **um** expecting]* **well** we should have higher expectations than that." But, as in the example in the preceding footnote, the annotation tool cannot output a reparandum with multiple IP's.

[12] http://macears.ll.mit.edu/macears_docs/data/SimpleMDE_V*x.y*.pdf — where *x* and *y* indicate the version.

[13] Reflecting the "clean up" orientation, the EARS RT-03 metadata model introduced the neologism "*DEPOD*", defined as the DEletable Part Of a Disfluency. This what we are now calling the *deletable region*.

[14] Note that filler disfluencies and edit disfluencies (and their deletable regions) could be defined as words rather than regions of time. The RT-04F submissions are, however, in terms of time.

[15] For example, a 1-hour news broadcast processed in 10 hours is counted as 10 times realtime regardless of whether the broadcast is stereo or monaural. And a 5-minute telephone conversation processed in 50 minutes would also count as 10X realtime, whether the signal is a 4-wire/2-channel signal or a 2-wire/1-channel signal.

(cough, sneeze, breath, lipsmack, and laugh), or non-speech sounds (such as door slams and so forth).

The RT-04F STT evaluation will include data sets in English, Arabic, and Mandarin.

TOKEN STRING FORMATTING

A single standardized spelling is required for scoreable lexemes, and the STT system must output this spelling in order to be scored as correct.**[16]** Homophones must be spelled correctly according to the given context in order to be considered correct. All tokens are to be generated according to Standard Normal Orthographic Representation (SNOR) rules:

Whitespace-separated lexical tokens (for languages that use whitespace-defined words)

Case insensitive alphabetic text (usually in all upper case)

Spelled letters are represented with the letter followed by a period (e.g., "a. b. c.")

No non-alphabetic characters (except apostrophes for contractions and possessives and hyphens for hyphenated words and fragments)

Note that in scoring, hyphenated words will be divided into their constituent parts. Thus, for scoring, a hyphen within a token will be treated as a token separator. A hyphen at either end of a token string indicates the missing part of a spoken fragment.

3.3 STT EVALUATION FRAMEWORK

The STT task is similar to previous ASR "Hub-4" and "Hub-5" evaluations, but with additions to support the classification of output tokens and (optionally) speaker assignment. The existing scoring conventions will be used unchanged from RT-03S.

The STT performance measure is essentially the same as the traditional NIST ASR WER measure using the NIST SCLITE software. The primary metric for the RT-04F STT evaluation will (as in RT-03S) be calculated over non-overlapping speech (i.e., omitting regions with multiple reference speakers in the same channel speaking simultaneously). [17]

3.3.1 SYSTEM OUTPUT GENERATION

The system output will be a CTM[18] file (see section 10.2.2). A CTM file is token-based and is to include the following information for each recognized token: the name of the source file, the channel processed, the beginning time of the recognized token, the duration of the recognized token, the string representation of the recognized token, a confidence probability, a token type, and a speaker identifier. The speaker information is optional, but is included to support STT/MDE fusion experiments. If no speaker information is generated, a value of "unknown" should be used for lexical token types and "null" for non-lexical token types. See section 10.2.2 for specific formatting requirements. The following describes each possible system output (CTM) token type[19]:

lex - a lexical token.

frag - a lexical fragment.

Note: An optional hyphen may also be used in the token string to indicate the missing (unspoken) part of the token, but the **frag** type must also be used.

fp - a filled pause.

un-lex - an uncertain lexical token. This type tag is normally used only in the reference.

for-lex - a "foreign" lexical token. This type tag is normally used only in the reference.

non-lex - a non-lexical acoustic phenomenon (breath-noise, door-bang, etc.)[20].

misc - other annotations not covered in above.**[21]**

Of the token types listed above, all types other than **lex** will be stripped from the system output prior to STT scoring, and in the reference they will be tagged as "optionally deletable". Therefore only tokens tagged as type lex in the system output will be aligned and scored, and all others (because stripped out) may be regarded as optional. Although systems aren't penalized (or rewarded) for outputting those optional types, we encourage their output to support metadata experiments.

3.3.2 REFERENCE TOKEN PROCESSING

A Segment Time Marked (STM) scoring reference is generated from the human reference transcripts.[22] Contraction expansions are annotated in the human reference: the annotator will choose (and the STM file will contain) the single most likely expansion for each contraction. Non-scoreable regions (such as untranscribed areas) are explicitly tagged in the STM file for exclusion from scoring (there will be no scoring UEM file for the STT evaluation). The tokens of the various STM token types[19] in the STM reference will be processed as follows:

lex – STM tokens of type **lex** are not specially tagged in the reference. As such, they are aligned and scored.

fp – STM tokens of this pause-filler type are tagged as optionally deletable[23] in the reference. As the first step in

[16] Token spelling is determined by NIST by first consulting an authoritative reference – e.g., the American Heritage Dictionary (AHD) for English. Lacking an authoritative reference, the www is searched to find the most common representation. If no single form is dominant, then two or more forms will be permitted via an orthographic map file. As in previous years, a transcription filter and orthographic map file will be used on both the reference and hypothesis transcripts to apply rules for mapping common alternate representations to a single scoreable form.

[17] Note that anticipated upcoming domains in future evaluations, such as STT transcription of meetings, will include processing of overlapping speech.

[18] The CTM file format is one of the immediate predecessors of the RTTM file format. The CTM and RTTM file formats *differ*.

[19] Note that in the RTTM format, some of what are token types in CTM and STM format data are instead subtypes of the RTTM *lexeme* type.

[20] RTTM (the reference data for the MDE evaluations) divides this category into non-speech (non-vocal noises) and non-lex (vocal noises). See Appendix A.

[21] A system may give this tag to any token which is to be excluded from scoring – including tokens for which the more specific CTM types exist. But where possible, sites are encouraged to use the supported more specific CTM types to enhance the usefulness of the data for MDE experiments.

[22] See ftp://jaguar.ncsl.nist.gov/current_docs/sctk/doc/infmts.htm

[23] "Optionally deletable" means that a system may omit the token without penalty, but if the system does output the token then it will be scored as correct or incorrect. Optionally deletable tokens contribute to the count of reference tokens (the WER denominator) whether or not the system outputs them.

scoring them, these tokens in the system output will be replaced by a generic internal fp token. Their orthography will be ignored.

frag - STM tokens of type **frag** are tagged in the reference both as optionally deletable and as fragments. They contribute to the WER denominator. Note: In addition, if a system output token of type lex aligns with a frag in the reference, it is counted as correct if the reference frag token string is a substring of the system output token string.[24]

un-lex, for-lex – Tokens of these types are tagged as optionally deletable in the reference. They contribute to the WER denominator.

non-lex and **misc** – These token types are removed from the reference

3.3.3 GLM PROCESSING

Prior to scoring, both the reference and system output token strings will be transformed using a global map file (GLM). The GLM is intended to ensure that reference and hypothesis tokens which do not differ semantically are scored as correct. This is accomplished by transforming the token strings in both the reference and system output via a set of mapping rules. The GLM applies a set of rules to the system output which expands contractions to all possible expanded forms.

Note that GLM processing may result in the generation of several alternative token strings in the system output. It may also result in token strings being split into two or more strings. For example, contractions are mapped to their expanded form and compound words are split into their constituents. After GLM filtering, hyphens in both the system output and reference are transformed into token separators.

3.3.4 SCORING

Once the pre-processing is complete, token alignment will be performed using a token-mediated alignment optimized for minimum word error rate.

3.4 STT EVALUATION METRICS

An overall STT error score will be computed as the average number of token recognition errors per reference token:

$$Error_{STT} = \left(N_{Del} + N_{Ins} + N_{Subst}\right) / N_{Ref}$$

where

N_{Del} = the number of unmapped reference tokens,

N_{Ins} = the number of unmapped STT output tokens,

N_{Subst} = the number of mapped STT output tokens with non-matching reference spelling per the token rules above, and

N_{Ref} = the maximum number of reference tokens[25]

As an additional optional performance measure, the confidence of a system in its transcription output will be evaluated. In order to do this, the system must attach a measure of confidence to each of its scoreable output tokens. This confidence measure represents the system's estimate of the probability that the output token is correct and must have a value between 0 and 1 inclusive. The performance of this confidence measure will be evaluated using the same normalized cross entropy score that NIST has been using in previous ASR evaluations.[26]

Conditioned Sub-Scoring:

STT WER performance statistics will be tabulated for the following conditions:

Language – Performance will be measured separately for English, Chinese (Mandarin), and Arabic language data.

Source – Performance will be measured separately for broadcast news sources and for telephone conversations.

CPU processing time – See section 3.1 and Appendix B for processing time options, calculation, and requirements.

Speaking conditions – Performance will be measured separately for the following speaking conditions:

Non-overlapping speech. (primary metric for EARS)

Overlapping speech

All speech

4 THE RT-04F STRUCTURAL METADATA TASKS

RT-04F features a variety of tasks related to metadata that are each being evaluated.

Metadata extraction (MDE)

Structural metadata

- Edit Word Detection (EWD)
- Filler Word Detection (FWD)
- IP Detection (IPD)
- SU Boundary Detection (SUBD)

Diarization

- Who spoke when

This section (section 4) of the document deals with the structural-metadata extraction tasks. The following section (section 5) deals with the "who spoke when" diarization metadata extraction task.

The EWD and FWD tasks require the system to specify regions of time, but their primary metrics are word-based and are computed by determining which words are covered[27] by the regions of time that the systems have identified.

The IPD and SUBD tasks require the systems to specify points of time, and the SUBD task also requires the system to identify the type of each SU. Their primary metrics are detection based (detection includes getting the type correct for SU's).

The "who spoke when" task requires the system to identify regions of time and can be performed without the system generating (or submitting) any STT output at all. The primary metric is time-based.

[24] But not the other way round. A complete word in the reference will never align to a frag in the system output because all frag's in the system output get stripped out before alignment occurs.

[25] N_{Ref} includes all scoreable reference tokens (including optionally deletable tokens) and counts the maximum number of tokens (e.g., the expanded version of contractions). Note that N_{Ref} considers only the reference transcript and is not affected by tokens in the system output transcript, regardless of their type.

[26] http://www.nist.gov/speech/tests/rt/rt2003/doc/NCE.htm

[27] A word is covered by a region of time if the mid-point time of the word falls within the region of time.

In contrast, rather than specifying times, the SASTT task requires the system to specify the words (STT) and to attribute a speaker (speaker label) to each.

The RT-04F metadata tasks other than "who spoke when" all require the systems to have (or generate as STT output) a list of the words in the speech signal.

The system output for the RT-04F metadata tasks will be submitted in RTTM format.

4.1 SCOREABLE STRUCTURAL METADATA TOKENS

The structural metadata and integrated-task systems that are being evaluated produce sequences of tokens to represent acoustic events in the speech signal. Such token sequences can be used for two purposes. First, they can be used to align the system output with the reference. Second, they can be used to measure the accuracy of the system output against the reference.

Systems will submit RTTM format data[28] for all the RT-04F metadata tasks Note that in the RTTM format, some of what are token types in CTM and STM format data are instead subtypes of the RTTM *lexeme* type.

In RTTM format submissions, tokens of type *lexeme* will be aligned and scored. In the case of **lex** and **for-lex** subtypes of *lexeme* tokens, identical, un-cased orthography matches between the reference and system outputs will constitute a correct match during the token alignment and token scoring process (see GLM processing in section 3.3.3 for additional information). In the case of the **frag** and **fp** subtypes of *lexeme* tokens, if the subtypes match, the tokens are considered a correct match during the token alignment and token scoring process even when the orthographies do not match.

Certain tokens can occur in reference token sequences but will never fall within an evaluable region of the evaluation data. There is one such *lexeme* token subtype:

- **un-lex** – a representation of a word whose identity is not clear to the human transcriber, or words infected with or affected by laughter.

4.2 STRUCTURAL-METADATA EXTRACTION FRAMEWORK

Systems are given only a digital audio signal as input. Some of the tasks are defined in terms of detection of "extent", i.e., the system must detect and output one or more spans indicating the locations and durations of particular metadata events. Others tasks require the detection of "points", i.e., the system must detect and output events that occur at a particular instant in time. A system may implement any combination of the tasks.

For RT-04, the NIST metadata extraction framework defines four structural-metadata detection tasks (for Edit Words, Filler Words, Interruption Points, and SU Boundaries), one integrated task (Speaker-Attributed STT), and one placeholder task pending any replacement proposals[29] (04RT).

Except for the "who spoke when" diarization task, STT output will be required from each system to allow for alignment between reference and system hypotheses and for the congruence of STT and metadata events.[30]

STT output will be in the form of a sequence of tokens. The start times and durations for each such token will be needed for the MDE scoring process.

All reference data will be distributed as RTTM files, UEM files, and the relevant GLM file. All submissions of system output for MDE scoring (including for who spoke when) shall be in RTTM format, and no other data format will be accepted.

Two UEM-formatted files (see section 7) will be used. The metadata *scoring* UEM file will exclude non-transcribed regions (commercials are among the non-transcribed material). The metadata *input* UEM file will identify the entire broadcast to be processed. This input UEM file will *not* exclude commercials.

4.2.1 REGIONS IGNORED BY METADATA SCORING SOFTWARE

In addition to regions excluded by the scoring UEM file, command-line options in the scoring software will cause the scoring software to exclude the following regions from scoring.

1. Overlapping speech: These regions of time include speech from multiple speakers in the same channel.
2. Unannotated SUs: SUs can have the type "unannotated" if the LDC was unable to perform SU annotation on that stretch of speech.
3. Unannotated Metadata Regions: Within the transcript, any regions marked with the NO_RT_METADATA annotation.
4. Any SEGMENT for which the speaker is **<NA>**. (Note that the fundamental reference data is missing for these segments, so that they are effectively non-transcribed).

4.2.2 SUMMARY OF EVALUATION PROCEDURE

The evaluation procedure consists of three stages. First, the scoreable token sequences from the reference and system output are aligned (using Dynamic Programming) to compute the minimum Edit Distance[31] between the two token sequences (edit distance is usually called the Levenshtein Distance, after the paper[32] by V. I. Levenshtein that appears to have introduced the idea). This alignment is fixed for the remainder of the evaluation procedure.

Second, additional separate mappings of metadata are performed to support metadata detection metrics[33] and

[28] Thus, if a system's STT output is in CTM format, the system must convert that data to RTTM before submitting it for the MDE evaluations.

[29] The program sponsor has not expressed significant interest in a 04rt token-error-rate (TER) metric that is just a minimal update of the 03rt comprehensive Rich Transcription task from RT-03F.

[30] For diagnostic purposes, performance will also be reported without applying this STT-based alignment.

[31] Edit Distance is the minimum number of edits (insertions, deletions, and substitutions) necessary to convert one string into another. The three kinds of edits are simply counted (in effect, equally weighted).

[32] V. I. Levenshtein: "*Binary Codes Capable of Correcting Deletions, Insertions and Reversals*", in Soviet Physics Doklady, Vol. 10, Nr. 8, Feb. 1966, pp. 707 – 710.

[33] This is necessary because the scoring of the metrics allows the system to split or merge disfluencies. For example, when a system filler disfluency spans multiple reference fillers, the optimal mapping (best score) is not found by looking at a single reference filler disfluency at a time.

speaker attribution metrics. Finding the minimum-error mapping of the system speaker labels to the reference speaker labels is a Bipartite Graph Matching problem. After the optimal mapping is determined, the speaker labels on the system output tokens are, in effect, changed into their mapped reference equivalents.

In the third stage, an error rate is calculated for each of the RT-04F metadata tasks.

All metrics are defined over the alignment produced in the first stage. This common alignment operates on the set of scoreable tokens as defined in Section 4.1. In computing the Edit Distance between the reference and system output token sequences

- lexeme tokens of any of the four scoreable lexeme subtypes are allowed to align to any other type if the orthography matches,
- lexeme and foreign-lexeme tokens are considered matched if their un-cased orthographic representations are the same, and
- when a system token and a reference token are both of type *filled-pause* or both of type *fragment*, they are matched based on their type only without regard to their orthography.

While the common alignment is governed principally by the token orthography and type, metadata (expressed as token subtypes or attributes) also exerts an influence upon the alignment whenever the orthographies differ between the tokens being compared. In other words, metadata is not permitted to dislodge a token from an alignment that results in an orthographic or type match, but wherever the orthographies or types are mismatched, the alignment is optimized jointly for all the metadata. This can be implemented as a simple table of substitution weights used in computing the Edit Distance.

For calibration purposes, we also compute a Word Error Rate (corresponding to that computed by SCLite) for the common alignment based upon the orthographic and type matches only, disregarding the metadata attributes.

4.3 Structural MDE Evaluation Metrics

Separate performance measures are defined for each of the EARS MDE tasks. For each, the number of errors is accumulated over all of the files and channels then normalized into one average for the system on that task.

4.3.1 Conditioned Sub-Scoring

MDE performance statistics will be tabulated separately for each of the four combinations of data source (broadcast news sources or telephone conversation) and input condition. (speech-plus-reference or speech-only). The input conditions are described in section 9.

4.3.2 Edit Word Detection

The Edit Word Detection (EWD) task is to detect regions of the input signal containing the words in *deletable regions* of edit disfluencies, as they are defined in SimpleMDE Annotation Specification. For the RT-04F evaluation, the detection task requires the system to specify the start time and duration of the deletable regions. The scoring will be in terms of the words covered by these regions of time.

There is no reward or penalty for splitting a single detected region into two or more contiguous regions having identical overall extent. Nor is there any reward or penalty for combining two or more contiguous detected regions into a single detected region of identical extent.

An edit disfluency may have fillers that occur within its deletable region. For the purposes of the Edit Word Detection task, regions containing such filler tokens should be detected as part of this task.

For the RT-04F evaluation, automatic identification of edit disfluency subtype is *not* part of the Edit Word Detection task.

The primary metric is as follows.

$$Error_{EditWordDetection} = \frac{\left(\begin{array}{l} \#\text{ of deletable ref edit tokens that are} \\ \qquad \text{not covered by deletable regions of sys edits} \\ +\#\text{ of other ref tokens} \\ \qquad \text{covered by deletable regions of sys edits} \end{array}\right)}{\#\text{ of deletable ref edit tokens}}$$

In addition, the software will output each of the three components of the metric (the denominator and the two terms of the numerator).

The formula refers to deletable edit tokens, which means tokens that are covered by the deletable regions of edit disfluencies. A token is “covered” by a deletable region if the midpoint (i.e., the average of beg time and end time) of the token falls within that deletable region’s time interval.

4.3.3 Filler Word Detection

The Filler Word Detection (FWD) task is to detect regions of the input signal containing fillers and to correctly detect the subtype of fillers. Fillers are defined in the SimpleMDE Annotation Specification. This detection task requires the system to specify the start and duration of all regions of the input signal containing fillers and to specify the subtype of each, (filled-pause, discourse marker, or explicit editing term).

In the primary metric for FWD, there is no reward or penalty for splitting a single detected region into two or more contiguous regions having identical overall extent. Nor is there any reward or penalty for combining two or more contiguous detected regions into a single detected region of identical extent.

Filler tokens may occur within the reparandum (as well as editing phase) of an edit disfluency, and for the purposes of the Filler Word Detection task, these tokens should be detected as part of this task. Thus, each such filler token should be detected in the Edit Word Detection task and also in the Filler Word Detection task.

Section 2 of the Simple Metadata Annotation Specification defines four subtypes of fillers (filled-pauses, discourse markers, explicit editing terms, asides/parentheticals). For the purposes of the Filler Word Detection task, regions containing fillers of subtype “aside/parenthetical” should *not* be detected.

The primary metric is as follows.

$$Error_{TypedFillerWordDetection} = \frac{\left(\begin{array}{l} \text{\# of ref filler tokens that are} \\ \qquad \text{not covered by sys fillers} \\ + \text{\# of ref filler tokens that are} \\ \qquad \text{covered by sys fillers of a different subtype} \\ + \text{\# of non - filler ref tokens that are} \\ \qquad \text{covered by sys fillers} \end{array} \right)}{\text{\# of ref tokens in the ref fillers}}$$

In addition, the software will output each of the four components of the metric (the denominator and the three terms of the numerator).

A token is "covered" by a filler if the midpoint (i.e., the average of beg time and end time) of the token falls within the filler's time interval.

4.3.4 IP DETECTION

The Interruption Point Detection (IPD) task is to produce the locations in time where interruption points occur. Interruption points are discussed in the Simple Metadata Annotation Specification (see footnote 2 and section 3.2 of that document). For the RT-04F evaluation, the detection task requires the system to specify the location in time of each interruption point. A complex edit disfluency will have multiple interruption points.

An interruption point (IP) occurs at the [right-hand] end of the deletable region of an edit disfluency (the reparandum in the case of a simple edit), which may be followed by a filler (e.g., a non-empty editing phase will be a filler and will be separately marked as a filler). And an IP occurs at the beginning of a filler. So, the deletable region of an edit followed by a filler suggests two contiguous IPs (one for the end of the deletable region and one for the beginning of the filler). In this situation, systems should output either one or two IPs according to the following rule.

> When a filler follows the deletable region of an edit within the same SU (i.e., not separated by an incomplete or complete SU boundary) and when there are no intervening RTTM tokens of type "lexeme" (see Appendix A) between the deletable region of the edit and the filler, a single, shared IP should be output. The location of such a shared IP should be specified as the time of the end of the deletable region of the edit. This sharing is independent of the gap in time between the end of the deletable region of the edit and the beginning of the filler. If these conditions are not met, two IPs should be emitted.

For the RT-04F evaluation, automatic identification of IP subtype is *not* part of the IP detection task.

The overall IP error rate will be simply the average number of missed IP detections and falsely detected IPs per reference IP:

$$Error_{IP} = \frac{\left(\text{\# of missed IP's} + \text{\# of false alarm IP's}\right)}{\text{\# of ref IP's}}$$

In addition, the software will output each of the three components of the metric (the denominator and the two terms of the numerator).

4.3.5 SU BOUNDARY DETECTION

The SU Boundary Detection task is to detect SU endpoints[34] and each SU's subtype. The definition of an SU[35] is provided in the SimpleMDE Annotation Specification. For the RT-04F evaluation, this task requires the system to specify the start time of the SU and its duration (from which the scoring software calculates its endpoint time). The system must also identify the SU's subtype (statement, question, backchannel, or incomplete).

The primary overall SU error score will be computed as the average number of missed SU end point detections, falsely detected SU end points, and correctly detected SU end points that are incorrectly classified as the wrong SU subtype, per reference SU:

$$Error_{\text{typed SU Detection}} = \frac{\left(\begin{array}{l} \quad \text{\# of missed SU end points} \\ + \ \text{\# of false alarm SU end points} \\ + \ \text{\# of detected SU end points with subtype incorrect} \end{array} \right)}{\text{\# of ref SU end points}}$$

In addition, the software will output each of the four components of the metric (the denominator and the three terms of the numerator).

5 DIARIZATION – "WHO SPOKE WHEN" MDE

A transcript where the speakers are labeled, so that the reader can tell who spoke when, is more readily interpreted. This RT-04F MDE task will be like the RT-03S speaker segmentation "who spoke when" evaluation, except that the task will only be performed on Broadcast News[44] datasets and non-speech regions will be excluded from scoring by the scoring software rather than via a UEM file.

Diarization is the process of annotating an input audio channel with information that attributes (possibly overlapping) temporal regions of signal energy to their specific sources. These sources can include particular speakers, music, background noise sources, and other signal source/channel characteristics.

For RT-04, diarization will be limited to just the speaker segmentation "who spoke when" task, including speaker type (gender) classification. For the "who spoke when" task, small pauses in a speaker's speech, of less than 0.3 seconds, are not considered to be segmentation breaks. Material containing no pauses of 0.3 seconds or more should be bridged into a single continuous segment. Although somewhat arbitrary, the cutoff value of 0.3 seconds has been determined to be a good approximation of the minimum duration for a pause in speech

[34] One per SU. Thus this amounts to SU detection.

[35] SUs have been variously called "slash units", "sentence units", "sentence-like units", "semantic units" and "structural units".

resulting in an utterance boundary. Systems should consider vocal noise (laugh, cough, sneeze, breath, lipsmack) to be silence in constructing segment boundaries.[36] Systems are to identify the speaker type: adult_male, adult_female, child, or unknown. These speaker-type labels must be consistently applied to all segments attributed to a particular speaker[37].

For RT-04, the data to be processed for the “who spoke when” task will be the Broadcast News data sets.

Although many systems perform the diarization task without transcribing the text, note that systems may make use of the output of a word/token recognizer (or any other form of automatic signal processing) in performing this task. The approach used should be clearly documented in the task system description.

5.1 Speaker Segmentation Diarization Scoring

In order to measure performance, an optimum one-to-one mapping of reference speaker IDs to system output speaker IDs will be computed. The measure of optimality will be the aggregation, over all reference speakers, of time that is jointly attributed to both the reference speaker and the (corresponding) system output speaker to which that reference speaker is mapped. This will always be computed over all speech, including regions of overlap[38]. Mapping is subject to the following restrictions:

- Each reference speaker will map to at most one system output speaker, and each system output speaker will map to at most one reference speaker. If the system performance is perfect, this mapping will be one-to-one.
- Mapping of speakers will be computed separately for each speech data file.

Although the speaker mapping will take regions of overlapping speech into account, the primary metric will be based on non-overlapping speech only.

In addition, since segment times are assumed to be correct in the reference in this evaluation, no time collars will be employed to forgive timing errors in the reference.

Speaker detection performance will be expressed in terms of the miss and false alarm rates that result from the mapping.

An overall time-based speaker diarization error score will be computed as the fraction of speaker time that is not attributed correctly to a speaker. This will be the **primary metric** for speaker segmentation diarization:

$$Error_{SpkrSeg} = \frac{\sum\limits_{\substack{\text{all}\\\text{segs}}} \left\{ \mathrm{dur}(seg) \cdot \left(\max\left(\mathrm{N}_{\mathrm{Ref}}(seg), \mathrm{N}_{\mathrm{Sys}}(seg) \right) - \mathrm{N}_{\mathrm{Correct}}(seg) \right) \right\}}{\sum\limits_{\substack{\text{all}\\\text{segs}}} \left\{ \mathrm{dur}(seg) \cdot \mathrm{N}_{\mathrm{Ref}}(seg) \right\}}$$

where the speech data file is divided into contiguous segments at all speaker change points[39] and where, for each segment, *seg*:

$$\begin{aligned} \mathrm{dur}(seg) &= \text{the duration of } seg, \\ \mathrm{N}_{\mathrm{Ref}}(seg) &= \text{the \# of reference speakers speaking in } seg, \\ \mathrm{N}_{\mathrm{Sys}}(seg) &= \text{the \# of system speakers speaking in } seg, \\ \mathrm{N}_{\mathrm{Correct}}(seg) &= \text{the \# of reference speakers speaking in } seg \\ &\quad\ \text{for whom their matching (mapped) system} \\ &\quad\ \text{speakers are also speaking in } seg. \end{aligned}$$

The numerator of the overall diarization error score represents speaker diarization error time, and it can be decomposed into speaker time that is attributed to the wrong speaker, missed speaker time, and false alarm speaker time.

Speaker time that is attributed to the wrong speaker (called speaker error time) is the sum of the following over all segments:

$$\mathrm{dur}(seg)^* \ \{\min(\mathrm{N}_{\mathrm{Ref}}(seg),\ \mathrm{N}_{\mathrm{Sys}}(seg)) - \mathrm{N}_{\mathrm{Correct}}(seg)\}.$$

Missed speaker time is the sum of the following over only segments where more reference speakers than system speakers are speaking:

$$\mathrm{dur}(seg)^*(\mathrm{N}_{\mathrm{Ref}}(seg) - \mathrm{N}_{\mathrm{Sys}}(seg)).$$

False alarm speaker time is the sum of the following over only segments where more system speakers than reference speakers are speaking:

$$\mathrm{dur}(seg)^*(\mathrm{N}_{\mathrm{Sys}}(seg) - \mathrm{N}_{\mathrm{Ref}}(seg)).$$

No segment is both miss time and false-alarm time.

Word-based counterparts to the time-based speaker diarization error score, and to each of its three parts (speaker error time, missed speaker time, false-alarm speaker time), are also calculated and reported — by using word counts instead of time. These word-based versions count the number of reference words covered by the segment (a word is covered by a segment if the word’s midpoint time[40] falls in the segment. (midpoint time is the start time of the word plus half its duration).

In areas of overlap (segments where more than one reference speaker is speaking), note that the duration of the segment is attributed to all the reference speakers who are speaking in the segment, thus counting the time more than once. But since the reference data tells us which speaker actually spoke each reference word, we can (and do) attribute each word to its actual speaker, and in areas of overlap this means the words are not counted more than once.

A system may, optionally, attach a measure of confidence to each of its output speaker segments. This confidence measure represents the system’s estimate of the probability that the speaker of this segment is correctly assigned.[41] This confidence measure will not, however, be evaluated.

Using this optimal mapping of reference speaker IDs to system speaker IDs, the scoring software will also compute and report the accuracy of recognition of speaker type (adult_male, adult_female, child, or unknown). There will be two versions of this speaker-attribute-mapping information: one over just

[36] However, special scoring rules will apply to areas containing vocal noise. See Section 5.

[37] No sex change in mid conversation.

[38] By “overlap” we mean regions where more than one reference speaker is speaking on the same audio channel.

[39] A “speaker change point” occurs each time any reference speaker or system speaker starts speaking or stops speaking. Thus, the set of currently-speaking reference speakers and/or system speakers does not change during any segment.

[40] Midpoint time is the average of the start time and the end time.

[41] The confidence measure represents the confidence in speaker assignment only. It should exclude consideration of the correctness of other attributes such as speaker type and segment times.

successfully detected speakers (i.e., for mapped speakers) and the other (separately) over all system output speakers. The primary metrics, however, for the speaker-type diarization task are described in the next section.

5.2 Speaker Type (Gender) Diarization Scoring

The diarization "who spoke when" scoring program can be run in a mode that uses the speaker type (adult_male, adult_female, child, or unknown) as the speaker ID. In this mode, the program will bypass the algorithm to compute an optimum mapping of reference speakers to system output speakers, as the correct mapping (e.g., adult_female to adult_female) is known *a-priori.* As a result, more of the time and words are likely to be mapped than when the mapping was based on speaker IDs. The output in this mode will include the same time-based and word-based metrics described above, but will also include confusion matrices for the speaker types.[42] Using the speaker type as the speaker ID, the primary metric for speaker type diarization is calculated the same as indicated above for speaker segmentation diarization.

5.3 Speaker-weighted Diarization Scores

The SpkrSegEval software also calculates a proposed speaker-weighted who-spoke-when diarization-error metric[43]. This metric will continue to be calculated in order to further explore the behavior of the proposed metric. It is not, however, part of the official metric set for RT-04.

5.4 Conditioned Sub-Scoring

MDE Who Spoke When Diarization segmentation statistics will be tabulated separately for by Speaker ID and by Speaker Type (gender).

6 Integrated STT and Metadata tasks

There will be two integrated tasks: Speaker Attributed STT (SASTT) and a Rich Transcription rolled-up metric (04rt).

6.1 Speaker Attributed STT (SASTT)

The Speaker Attributed STT task is to produce a sequence of scoreable tokens and to identify the speaker for each token in a Broadcast News recording[44]. By "identify", we mean that the system must make an N-way decision as to the identity of the speaker of each token. "N" is the total number of speakers within a single input audio signal. "N" is not known to the system. All tokens spoken by the same speaker should be given the same, but arbitrary, speaker identification label. Tokens spoken by different speakers should be assigned different speaker identification labels. For the RT-04F evaluation, automatic identification of the proper name of the speaker is not part of the Speaker Attributed STT task.

A speaker-attributed token is any scoreable token (lexeme, foreign-lexeme, fragment, or filled-pause). A speaker attributed token is correct if (1) it counts as correct[45] for the STT metric[46] and (2) the scoring software mapped the speaker label on the system token to the speaker label on the reference token (as described in the second paragraph of section 4.2.2). The primary metric for speaker-attributed STT (SASTT) will be

$$Error_{SASTT} = \frac{\left(\begin{array}{l} \text{\# of STT insertions} \\ + \text{ \# of STT deletions} \\ + \text{ \# of STT substitutions} \\ + \text{\# of tokens that are correct for STT but} \\ \qquad \text{with the speaker incorrect} \end{array}\right)}{\text{\# of ref tokens}}$$

In addition, the software will output each of the five components of the metric (the denominator and the four terms of the numerator).

6.2 04RT — Rich Transcription

> Editorial Note: This is essentially the 03 Rich Transcription (03rt) task (modified w.r.t. the metadata task changes). It has been left in as a place holder pending any new proposals for a combined RT task that do and do not include STT errors in their formulation. Please note the second paragraph of section 4.2 and its associated footnote (footnote number 29).

The RT-04 Rich Transcription placeholder task is to produce a sequence of scoreable tokens and for each token, to detect whether that token is covered by the deletable region of an edit disfluency, whether it is part of a filler and of the correct subtype, whether it is located at the end of an SU with the subtype correct, whether an interruption point occurs (before or after it), and to identify the speaker of the token.

RT-04 Rich Transcription is evaluated using Token Error Rate, which has the following form:

$$TER = \frac{100 * \ (\text{\# sub} + \text{\# del} + \text{\# ins})}{\text{\# ref tokens}}$$

# sub = number of reference tokens aligned to system tokens for which any of the following is true:

- the token does not count as correct for SASTT
- the reference token is a Filler token and the system token is not (or vice versa) or both are filler tokens but the subtypes do not match
- the reference token is an Edit token and the system token is not (or vice versa)

[42] These speaker type confusion matrices are always generated by the program, both for speaker segmentation scoring and speaker type scoring. However, they will differ for segmentation and type scoring since they are based on different mappings.

[43] See message to MACEARS from Greg Sanders on June 24, 2003, which explains the proposed metric in detail.

[44] Distinguishing the speakers on Conversational Telephone Speech (CTS) data amounts to speech activity detection (each speaker is on a separate channel) and is therefore not of separate interest as a SASTT or "who spoke when" diarization research task. SASTT (and "who spoke when" diarization) will *not* be evaluated on CTS datasets.

[45] STT errors for SASTT are determined from the same token alignment as is used for the other MDE tasks.

[46] This is a change: in the RT-03F evaluation, STT insertions and deletions were errors, but STT substitutions were not.

- the reference token is adjacent to one or two IPs and the system token is not (or vice versa)
- the reference token is an SU-boundary token and the system token is not (or vice versa) or both are SU-boundary tokens but the subtypes do not match

# del = number of reference Rich Transcription tokens for which there is no corresponding system Rich Transcription token (due to STT deletion)

# ins = number of system Rich Transcription tokens for which there is no corresponding reference Rich Transcription token (due to STT insertion)

# ref tokens = number of reference STT tokens

7 EVALUATION UN-PARTITIONED EVALUATIONS MAPS (UEM)

Un-partitioned evaluation maps (UEM)s are the mechanism the evaluation infrastructure uses to specify time regions within an audio recording. An *input* UEM file will be provided for all tasks (including STT), to indicate what audio data is to be processed by the systems. A *scoring* UEM file will be used to specify the time regions to be scored for all the RT-04F MDE tasks. No scoring UEM files will used in scoring the STT tasks (the STM files will be used to score the STT tasks).

7.1 UEM FILE STRUCTURE

The UEM file format is a concatenation of time mark records for a segment of audio in a speech waveform. The records are separated with a newline. Each record must have a file id, channel identifier [1 | 2], begin time, and end time. Each record follows this BNF format:

```
UEM :== <F><SP><C><SP><BT><SP><ET>
```

where,

`<SP>` indicates a space (“ “).

`<F>` indicates the file id, consisting of the path, filename, and extension of the waveform to be processed.

`<C>` indicates the waveform channel can have a value of "1" or "2".

`<BT>` indicates the beginning time of the segment measured in seconds from the beginning of the file which is time 0.

`<ET>` indicates the ending time of the segment measured in seconds from the beginning of the file which is time 0.

For example:

```
audio/dev/english/cts/sw_47620.sph 1 0 291.34
audio/dev/english/cts/sw_47621.sph 1 0 301.98
. . .
```

7.2 SYSTEM INPUT UEM FILES

A UEM file is provided with the evaluation data to define the regions of the audio that the system must process. The boundaries specified by the UEM file will include the beginning and end of a conversation or broadcast-news show.

7.3 METADATA SCORING UEM FILES

An MDE scoring UEM file is provided with the reference transcripts that defines the scoreable regions of the audio file. In addition to the boundaries specified by the system input UEM, the MDE scoring UEM excludes extended regions of non-transcribed speech. These extended untranscribed regions in the Broadcast News data for RT-04F will include commercial breaks, reporter chit-chat outside the context of a story, station identifications, promotions for upcoming broadcasts, public-service announcements, and long musical interludes.

The boundaries defined by the UEM file apply to all objects in the file: no word, speaker-turn, segment, or forced-aligned token can cross them.[47] As a result, re-running a forced-alignment process or running an alternative forced-alignment will not affect the UEM files.

8 CORPORA RESOURCES

To be determined.

9 EVALUATION CONDITIONS

There are many different conditions under which system performance may be evaluated. This section identifies those conditions for which performance will be computed and, of those, which are to be designated as the required evaluation conditions.

The following list of evaluation conditions apply to all RT-04F Evaluation tasks.

Data set:

Eval 04F

Progress (EARS STT contractors *only*)

Language:

English, (MDE tasks will be English-only in RT-04F)

Mandarin, and

Arabic

Domain:

Broadcast News (BN), and

Conversational Telephone Speech (CTS)

(Participants may build systems to address either or both of these domains, and may build a separate system for each of the two domains.)

Input:

Speech-only input. Any desired fully-automatic signal processing approaches may be employed (including the use of a site developed STT system). This is the required evaluation condition for Input for all RT-04F tasks.

Speech plus the reference transcriptions: The function of this evaluation condition (which only applies to MDE tasks) is to serve as a perfect-STT control condition. It is

[47] Boundaries that can be crossed by some object will be generated within the scoring software. Examples of such objects include regions of overlapping speakers, uncertain lexemes (un-lex), and regions surrounding non-lexeme or non-speech tokens. Further, regions that pertain to only part of the signal on a channel (for example, only one speaker) will also be handled by the scoring software rather than the UEM files.

an optional contrast evaluation condition. The system inputs will be RTTM formatted files derived from the reference RTTM files and placed in the 'input' directory (described in section 10.2.1 below) of the evaluation corpus. The derived RTTM files will contain only *lexeme* RTTM records — with the speaker's identity expunged, (replaced by <NA>), and with the lexeme subtypes *'alpha'*, *'acronym'*, *'interjection'*, *'propernoun*', and *'other'* mapped into the *lex* subtype.

All participants must agree to completely process all of the data for at least one task. This means that, at a minimum, the speech-input-only processing condition must be implemented.

10 PARTICIPATION INSTRUCTIONS

Participation is encouraged for all those who are interested in one or more of the RT-04F tasks. All participants must, however, agree to completely process all of the data for at least one task and must complete a required condition for that task. This means that, at a minimum, the speech-input-only processing condition must be implemented. Participants have the freedom to implement systems for either or both domains, Broadcast News or Conversational Telephone Speech.

All participating teams are required to submit a primary system on the required task-specific evaluation condition. Each team may only submit one primary system for each task. Any contrastive system submissions must have a corresponding primary system submission.

As a condition of participation, all sites must agree to make their submissions (system output, system description, and ancillary files) available for experimental use by other research sites. Further, submission of system output to NIST constitutes permission on the part of the site for NIST to publish scores and analyses for that data including explicit identification of the submitting site and system.

10.1 PROCESSING RULES

10.1.1 RULES THAT APPLY TO ALL EVALUATIONS

All developed systems must be fully automatic requiring no manual intervention to influence the system's decision-making infrastructure when generating the system output. Manual intervention is allowed to shepherd system processes but not to change any parameter settings or processing steps in response to knowledge or intuition gained from processing the evaluation data.[48]

The only exemption from the automatic processing restriction is for the reference text condition. Participants who use the reference text condition can manually add pronunciations to their dictionaries to enable forced alignment of the out-of-vocabulary items. Participants cannot use the lexical knowledge gained from the reference+speech-input system to modify their speech-input only system.

Systems will be provided with recorded SPHERE formatted waveform files and a UEM file specifying the speech files and regions within them to be processed. Each conversational telephone speech test waveform will be provided in 2-channel files, and both channels must be processed. Broadcast news speech test data will be presented in single channel files, one per broadcast.

While entire broadcast and conversation files will be distributed, only the material specified in the UEM test index file for the experiment to be run is to be processed. Material outside of the times specified in the UEM test index file is not to be used in any way (e.g., for adaptation).

10.1.2 ADDITIONAL RULES FOR PROCESSING BROADCAST NEWS

News-oriented material (audio, textual, etc.) generated during the preceding test epoch (February 2001) or after the beginning of the current test epoch (beginning December 1, 2003) **may not be used in any way for system development or training.** Broadcast news material must be processed in the chronological order of the date/time of the original broadcast. Although automatic adaptation may be performed using previously-processed material, systems may not "look ahead" in time at later recordings. Hence, processing must be complete on a particular broadcast news test file before moving on to the next file.[49] Any form of within-file adaptation, however, is permitted and systems may look backwards in time at previously-processed files. The show identity and original broadcast date are allowable side information that systems may use. Therefore, systems may make use of show-dependent models.

10.1.3 ADDITIONAL RULES FOR PROCESSING CONVERSATIONAL TELEPHONE SPEECH

Conversational telephone speech may be processed in any order and any form of automatic within-conversation and cross-conversation adaptation may be employed. No side information is provided for telephone conversations (e.g., corpus collection name, recording time, etc.). No manual or automatic segmentation will be provided, although systems may make use of segmentation outputs donated from other sites.

10.1.4 ADDITIONAL RULES FOR PERFORMING THE STT TASK

EARS contractors (and only EARS contractors) will process the Progress Test Set. The same system must be used to process both the Progress and Current Test sets.

Please note that to ensure the integrity of the Progress Test Set, special rules governing the use (and disposal) of this data must be strictly observed. These are specified in a document to be published at the EARS evaluation website at http://ears.ll.mit.edu/.

Note that all of the constraints specified for the English STT tests regarding training, adaptation, and processing also apply to the Non-English STT tests**.**

10.2 DATA FORMATS

10.2.1 AUDIO DATA AND OTHER CORRESPONDING INPUTS

For practicality, the recorded waveform files to be processed will be distributed on CD-ROM and the corresponding indices, annotations, and transcripts will be made available via the Web or FTP using an identical directory structure. After the evaluation, system outputs will be released in this structure as well.

[48] For example, after processing one file and before processing the next file, shepherding does not include doing anything to exploit knowledge gained *by the researchers* as a result of processing that file.

[49] This applies to *all* tasks.

Directory	Description
indices/	index files containing the list of files and times to be processed for particular experiments
audio/	audio files
input/<EXP-ID>/	ancillary data including reference annotations for various experiments – must be used in accordance with instructions for that experiment
output/<EXP-ID>/	system output submissions – will be made available as received for integration tests
reference/	reference transcripts and annotations for post-evaluation scoring and analyses

Note: EXP-ID specifies a unique identifier for each experiment and is defined in section 10.3.1.

For clarity, the "audio/" and "reference/" directories are subdivided into <DATA>/<LANG>/<TYPE> subdirectories:

where,

<DATA> is either [dev04|eval04]

<LANG> is one of [english | mandarin | arabic]

<TYPE> is either [bnews|cts]

The "indices/" directory contains a set of UEM test index files specifying the waveform data to be evaluated for each EXP-ID condition supported in this evaluation as described in 10.3.1 and these files are named <EXP-ID>.uem with the special site code "expt". Separate UEM files, defined in section 7, will be provided for each experiment for each supported <DATA>, <LANG>, and <TYPE>. Corresponding ancillary data for some control conditions is given in the "input/" directory under subdirectories with the same EXP-ID.

10.2.2 STT Output Format

The RT-04F STT output format will be the CTM format (.ctm filename extension), as in RT-03S. Each output file is to begin with two special comment lines specifying the experiment run and inputs used. These lines must appear at the beginning of the file and are to be formatted as follows:

The first line may be an optional special comment specifying the experiment ID as defined in section 10.3.1 (EXP-ID) and is of the form:

;; EXP-ID: <EXP-ID>

For example,

;;EXP-ID: bbn_03_stt10x_eval03_eng_cts_spch_1

If present, this optional special comment line must begin with two semicolons "`;;`". Note that for purposes of scoring, all lines beginning with two semicolons are considered comments and are ignored. Blank lines are also ignored.

The header comments are followed by a list of CTM records. See the list below for the specific supported token types.

The CTM file format is a concatenation of time mark records for each output token in each channel of a waveform. The records are separated with a newline. Each field in a record is delimited with whitespace. Therefore, field values may not include whitespace characters. Each record follows the following BNF format:

```
CTM-RECORD   :==   <SOURCE><SP><CHANNEL><SP>
<BEG-TIME><SP><DURATION><SP><TOKEN><SP>
<CONF><SP><TYPE><SP><SPEAKER><NEWLINE>
```

where

`<SP>` is whitespace.

`<SOURCE>` is the waveform basename (no pathnames or extensions should be included).

`<CHANNEL>` is the waveform channel: "1", "2", etc. This value will always be "1" for single-channel files.

`<BEG-TIME>` is the beginning time of the token. This time is a floating point number, expressed in seconds, measured from the start time of the file. [50]

`<DURATION>` is the duration of the token. This time is a floating point number, expressed in seconds. [50]

`<TOKEN>` is the orthographic representation of the recognized word/lexeme or acoustic phenomena. For English, this is represented as a string of ASCII characters. (a token in the context of a non-English test might be represented in Unicode or some other special character set.) Token strings are case insensitive and may contain only upper or lowercase alphabetic characters, hyphens (-), and apostrophes (') only. No special characters are to be included in this field to indicate the type of token. Rather, the "TYPE" field is to be used to indicate the token type. Note however that a hyphen may be used for fragments to indicate the missing/unspoken portion of the fragment. However, the "frag" TYPE must still be used.

`<CONF>` is the confidence score, a floating point number between 0 (no confidence) and 1 (certainty). A value of "NA" is used (in CTM format data) when no confidence is computed and in the reference data. [51]

`<TYPE>` is the token type. The legal values of <TYPE> are "lex", "frag", "fp", "un-lex", "for-lex", "non-lex", "misc", or "noscore". See Section 3 for details on generation and scoring rules for each of these types.

lex is a lexical token.

frag is a lexical fragment. Note: A (optional) hyphen may also be used in the token string to indicate the missing (unspoken) part of the token, but the frag TYPE must also be used.

[50] A required time accuracy for BEG-TIME and DURATION is not defined, but these times must provide sufficient resolution for the evaluation software to align tags with the proper token in the reference when time-alignment-based scoring is used. This alignment can be problematic in the case of quickly-articulated adjoining words. Therefore, systems should produce time tags with as much resolution as is reasonably possible. Note that the word with the shortest duration in the MDE development test set is 15 ms.

[51] STT systems are required to compute a confidence for each scoreable token output for this evaluation. The "NA" value may be used only for non-scoreable tokens.

fp is a filled pause.

un-lex is an uncertain lexical token normally used only in the reference.

for-lex is a “foreign” lexical token normally used only in the reference.

non-lex is a non-lexical acoustic phenomenon (breath-noise, door-bang, etc.)

misc is other annotations not covered above.[52]

noscore is a special tag used only in reference files for scoring to indicate tokens which should not be aligned or scored.

<SPEAKER> is a string identifier for the speaker who uttered the token. This should be “null” for non-speech tokens and “unknown” when the speaker has not been determined.

Included below is an example of STT system output:

```
7654 1 11.34 0.2 YES 0.763 lex 1
7654 1 12.00 0.34 YOU 0.384 lex 1
7654 1 13.30 0.5 C- 0.806 frag 1
7654 1 17.50 0.2 AS 0.537 lex 1
:
7654 2 1.34 0.2 I 0.763 lex 2
7654 2 2.00 0.34 CAN 0.384 lex 2
7654 2 3.40 0.5 ADD 0.806 lex 2
7654 2 3.70 .2 door-bang 0 non-lex null
7654 2 7.00 0.2 AS 0.537 lex 2
:
```

10.2.3 MDE OUTPUT FORMAT

The RT-04F data format, both for the reference data and for the system submissions, will be RTTM (with .rttm filename extension). See Appendix A for a description of the RTTM format. Each RTTM file corresponds to a single source file in the test.

10.2.4 SYSTEM DESCRIPTION

For each test run (for each unique EXP-ID), a description of the system (algorithms, data, configuration) used to produce the system output must be provided along with your system output. If multiple system runs are submitted for a particular experiment with different systems/configurations, explicitly designate one run as the primary system and the others as contrastive systems in the system description (as well as in the SYSID string in the submission filename). The system description information is to be provided in a file named:

<EXP-ID>.txt

(where EXP-ID is defined in Section 10.3.1)

and placed in the “output” directory alongside the similarly-named directories containing your system output. This file is to be formatted as follows:

1. EXP-ID = <EXP-ID>
2. Primary: yes | no
3. System Description:

   [brief technical description of your system; if a contrastive test, contrast with primary system description]

4. Training:

   [list of resources used for training; for STT, be sure to address acoustic and LM training, and lexicon]

5. References:

   [any pertinent references]

10.3 SUBMISSION INSTRUCTIONS

10.3.1 SUBMISSION EXPERIMENT CODES

The output of each submitted experiment must be identified by the following code as specified above.

EXP-ID =
<SITE>_<YEAR>_<TASK>_<DATA>_<LANG>_
<TYPE>_<COND >_<SYSID>_<RUN>

where,

SITE ::= expt | bbn | bbnplus | cu | elisa | clips | sri | sriplus | ibm | mitll | ms | pan | ...

(The special SITE code “expt” is used in the EXP-ID-based filename of the UEM test index files under the “indices/” directory to list the test material for a particular experiment and in the EXP-ID-based subdirectory name under the “input/” directory to indicate ancillary data to be used in certain control condition experiments.)

YEAR ::= 04

For the RT-04F Rich Transcription Evaluation, these are:

TASK ::= ewd | fwd | ipd | subd | sastt | 04rt | sttul | stt20x | stt10x | stt1x | sttulmb | stt10xmb | stt1xmb | data

where,

ewd = edit word detection

fwd = filler word detection

ipd = IP detection

subd = SU boundary detection

sastt = Speaker attributed STT

04rt = RT-04 rich transcription

sttul = STT with unlimited processing time

stt20x=STT running in less than or equal to 20 X realtime

stt10x=STT running in less than or equal to 10 X realtime

stt1x = STT running in less than or equal to 1 X realtime

sttulmb = STT with unlimited processing time, using a mothballed system

stt10xmb = STT running in less than or equal to 10 X realtime, using a mothballed system

stt1x = STT running in less than or equal to 1 X realtime, using a mothballed system

[52] Any token which is to be excluded from scoring may be given this tag – including those for which specified types exist. However, where possible, sites are encouraged to use the supported types to enhance the usefulness of the data for MDE experiments.

data = a special TASK code used to provide a directory for ancillary data such as common CTM files used over many MDE experiments. Please make sure to use increasing run numbers for this special experiment ID when making multiple submissions so that your ancillary data from earlier submissions is not over-written here at NIST

DATA ::= eval04f | prog

LANG ::= eng | man | arab

RT-04F STT will include all three languages

RT-04F MDE only includes English (eng) material.

TYPE ::= bnews | cts

CONDITION ::= spch | ref

where,

spch = audio input only

ref = audio input + reference transcript

The "spch" (speech) condition is the primary condition of interest. The "ref" (reference) condition is provided as a control for perfect speech recognition and includes both the speech and reference transcript as input.[53] The MDE tasks for this condition may make use of only the LEXEME entries in the supplied RTTM as defined in Section 9 "Evaluation Conditions".

SYSID ::= site-named string designating the system used

The SYSID string must be present. It is to begin with p- for a primary system or with c- for any contrastive systems. For example, this string could be p-wonderful or c-amazing.

This field is intended to differentiate between contrastive runs for the same condition. Therefore, a different SYSID should be created for runs where any manual changes were made to a particular system.

RUN ::= 1..n (with values greater than 1 indicating multiple runs of the same experiment/system)

An incremental run number ***must*** be used for multiple submissions of any particular experiment with an identical configuration (due to a bug or runtime problem.) This should *not* be used to indicate contrastive runs. Instead, a different SYSID should be used. However, please note that ***only*** the first run will be considered "official" and be scored by NIST unless special arrangements are made with NIST.

Please also note that submissions which reuse identical experiment IDs/run numbers from previous submissions will be automatically rejected.

Examples:

bbn_04_ip_eval04_eng_cts_spch_c-superreco1_1

sri_04_sastt_eval04_eng_bnews_ref_p-speakerid2_1

10.3.2 SUBMISSION DIRECTORY STRUCTURE

All system output submissions must be formatted according to the following directory structure:

output/<SYSTEM-DESCRIPTION-FILES>

output/<EXP-ID>/ <OUTPUT-FILES>

where,

<SYSTEM-DESCRIPTION-FILES> one per <EXP-ID> as specified in 10.2.4

<EXP-ID> is as defined in Section 10.3.1

<OUTPUT-FILES> are as in sections 10.2.2, section 10.2.3, and section 10.2.4.

Note: one output file must be generated for EACH input file as specified in the test index for the experiment being run.

The output files are to be named so as to be identical to the input file basenames with the appropriate .ctm or .rttm filetype extension. For example, an STT output file for the speech waveform file sw_47620.sph must be named sw_47620.ctm and an MDE output file must be named sw_47620.rttm.

When generated, these output files are to be placed under the appropriately-named EXP-ID directory on your system identifying the experiment run.

10.3.3 SUBMISSION PACKAGING AND UPLOADING

To prepare your submission, first create the previously-described file/directory structure. This structure may contain the output of multiple experiments, although you are free to submit one experiment at a time if you like. The following instructions assume that you are using the UNIX operating system. If you do not have access to UNIX utilities or ftp, please contact NIST to make alternate arrangements.

First change directory to the parent directory of your "output/" directory. Next, type the following command:

```
tar -cvf - ./output | gzip > <SITE>_<SUB-NUM>.tgz
```

where,

<SITE> is the ID for your site as given in section 10.3.1

<SUB-NUM> is an integer 1 – n where 1 identifies your first submission, 2 your second, and so forth.

This command creates a single tar file containing all of your results. Next, ftp to jaguar.ncsl.nist.gov giving the username 'anonymous' and your e-mail address as the password. After you are logged in, issue the following set of commands, (the prompt will be 'ftp>'):

```
ftp> cd incoming
ftp> binary
ftp> put <SITE>_<SUB-NUM>.tgz
ftp> quit
```

You've now submitted your recognition results to NIST. Note that because the "incoming" ftp directory (where you just ftp'd your submission) is write protected, you will not be able to overwrite any existing file by the same name (you will get an error message if you try) and you will not be able to list the incoming directory (i.e., with the "ls" or "dir" commands). So, pay attention to whether you get any error messages from the ftp process when you execute the ftp commands stated above.

The last thing you need to do is send an e-mail message to Audrey Le at audrey.le@nist.gov to notify NIST of your

[53] Reference-condition submissions are extremely useful for data analysis, so participants are encouraged to submit them.

submission. The following information should be included in your email:

1) The name of your submission file

2) A listing of each of your submitted experiment IDs

3) e.g.,

```
Submission: bbnplus_1 <NL>
Experiments: <NL>
bbnplus_04_subd_eval04_eng_cts_spch
_c-superreco1_1<NL>
bbnplus_03f_ipd_eval04_eng_cts_spch_c
-superreco2_1 <NL>
```

Please submit your files in time for us to deal with any transmission/formatting problems that might occur — well before the due date if possible.

Note that submissions received after the stated due dates *for any reason* will be marked late.

11 SCHEDULE

To be determined.

Please note that the stated dates are hard deadlines. All late submissions will be marked as such and given the tight schedule, severely late submissions may not be scored at all prior to the workshops.

12 WORKSHOPS

To be determined.

Information regarding workshop logistics and registration will be posted at a later date in email.

Appendix A: RTTM File Format Specification

We have renamed **propername** to **propernoun** and renamed **lip-smack** to **lipsmack**, to correspond to actual practice and actual reference data. There are four general object categories to be represented. They are STT objects, MDE objects, source (speaker) objects, and structural objects.[54] Each of these general categories may be represented by one or more types and subtypes, as shown in table 1.

Table 1 Rich Text object types and subtypes

Type	Subtypes
Structural types:	
SEGMENT	**eval**, or (none)
NOSCORE	(none)
NO_RT_METADATA	(none)
STT types:	
LEXEME	**lex**, **fp**, **frag**, **un-lex**[55], **for-lex**, **alpha**[56], **acronym**[56], **interjection**[56], **propernoun**[56], and **other**
NON-LEX	**laugh**, **breath**, **lipsmack**, **cough**, **sneeze**, and **other**
NON-SPEECH	**noise**, **music**, and **other**
MDE types:	
FILLER	**filled_pause**, **discourse_marker**, **explicit_editing_term**, and **other**
EDIT	**repetition**, **restart**, **revision**, **simple**, **complex**, and **other**
IP	**edit**, **filler**, **edit&filler**, and **other**
SU	**statement**, **backchannel**, **question**, **incomplete**, **unannotated**, and **other**
CB	**coordinating**, **clausal**, and **other**
A/P	(none)
SPEAKER	(none)
Source information:	
SPKR-INFO	**adult_male**, **adult_female**, **child**, and **unknown**

The STT, MDE and Source information objects are potential research target. And, except for the static speaker information object [**SPKR-INFO**], each object exhibits a temporal extent with a beginning time and a duration. (The duration of interruption points [**IP**] and clausal boundaries [**CB**] is zero by definition.)

These objects are represented individually, one object per record, using a flat record format with object attributes stored in white-space separated fields. The format is shown in table 2.

[54] Structural objects are important because they are produced by LDC to provide a modicum of temporal organization in the annotation and identify non-evaluable regions.

[55] Un-lex tags lexemes whose identity is uncertain and is also used to tag words that are infected with or affected by laughter.

[56] This subtype is an optional addition to the previous set of lexeme subtypes which is provided to supplement the interpretation of some lexemes.

Table 2 Object record format for EARS objects

Field 1	2	3	4	5	6	7	8	9
type	file	chnl	tbeg	tdur	ortho	stype	name	conf

where

file is the waveform file base name (i.e., without path names or extensions).

chnl is the waveform channel (e.g., "**1**" or "**2**").

tbeg is the beginning time of the object, in seconds, measured from the start time of the file.[57] If there is no beginning time, use tbeg = "**<NA>**".

tdur is the duration of the object, in seconds.[4] If there is no duration, use tdur = "**<NA>**".

stype is the subtype of the object. If there is no subtype, use stype = "**<NA>**".

ortho is the orthographic rendering (spelling) of the object for STT object types. If there is no orthographic representation, use ortho = "**<NA>**".

name is the name of the speaker. name must uniquely specify the speaker within the scope of the file. If name is not applicable or if no claim is being made as to the identity of the speaker, use name = "**<NA>**".

conf is the confidence (probability) that the object information is correct. If conf is not available, use conf = "**<NA>**".

This format, when specialized for the various object types, results in the different field patterns shown in table 3.

Table 3 Format specialization for specific object types

Field 1	2	3	4	5	6	7	8	9
Type	*File*	*chnl*	*tbeg*	*tdur*	*ortho*	*stype*	*name*	*conf*
SEGMENT	File	chnl	tbeg	tdur	**<NA>**	**eval** or **<NA>**	name or **<NA>**	conf or **<NA>**
NOSCORE	File	chnl	tbeg	tdur	**<NA>**	**<NA>**	**<NA>**	**<NA>**
NO_RT_METADATA	File	chnl	tbeg	tdur	**<NA>**	**<NA>**	**<NA>**	**<NA>**
LEXEME **NON-LEX**	File	chnl	tbeg	tdur	ortho or **<NA>**	stype	name	conf or **<NA>**
NON-SPEECH	File	chnl	tbeg	tdur	**<NA>**	stype	**<NA>**	conf or **<NA>**
FILLER **EDIT** **SU**	File	chnl	tbeg	Tdur	**<NA>**	stype	name	conf or **<NA>**
IP **CB**	File	chnl	tbeg	**<NA>**	**<NA>**	stype	name	conf or **<NA>**
A/P **SPEAKER**	File	chnl	tbeg	Tdur	**<NA>**	**<NA>**	name	conf or **<NA>**
SPKR-INFO	File	chnl	**<NA>**	**<NA>**	**<NA>**	stype	name	conf or **<NA>**

[57] If tbeg and tdur are "fake" times that serve only to synchronize events in time and that do not represent actual times, then these times should be tagged with a trailing asterisk (e.g., tbeg = **12.34*** rather than **12.34**).

Appendix B: Processing Time Calculation for System Descriptions

1. CTS Echo Cancellation

To keep the playing field level, you need not count echo cancellation in your realtime calculation. If you run it during recognition processing, the "official" realtime calculation you report should be (your total processing time, minus your echo cancellation processing time) divided by the recording duration.

2. RT-03S Processing Speed Computation — Total Processing Time (TPT):

For this and future RT evaluations, the time to be reported is the Total Processing Time (TPT) that it takes to process all channels of the recorded speech (including ALL I/O) on a single CPU.

TPT represents the time a system would take to process the recorded audio input and produce lexical token output as measured by a stopwatch.

So that research systems that aren't completely pipelined aren't penalized, the "stopwatch" may be stopped between (batch) processes.

Note that TPT should exclude time to implement CTS echo cancellation. This is so that sites using the Mississippi State Echo Cancellation Software, which was not optimized for speed or integration, are not penalized.

TPT may also exclude time to "warm up" the system prior to loading the test recordings (e.g., loading models into memory.)

Source Signal Duration (SSD):
In order to calculate the realtime factor, the duration of the source signal recording must be determined. The source signal duration (SSD) is the actual recording time for the audio used in the experiment as specified in the experiment's UEM files. This time is channel-independent and should be calculated across all
channels for multi-channel recordings.

Speed Factor (SF) Computation:
The speed factor (SF) (also known as "X" and "times-realtime") is calculated as follows:

```
   SF = TPT/SSD
```

For example, a 1-hour news broadcast processed in 10 hours would have a SF of 10 (regardless of whether the broadcast is stereo or monaural). And a 5-minute telephone conversation processed in 50 minutes would also have an SF of 10 (regardless of whether the signal is a 4-wire/2-channel signal or a 2-wire/1-channel signal).

Reporting Your Processing Speed Information:
Although we encourage you to break out your processing time components into as much detail as you like, you should minimally report the above information in the system description for each of your submitted experiments in the form:

```
    TPT = <FLOAT>
    SSD = <FLOAT>
    SF  = <FLOAT>
```